Лабораторна робота №4
Стиск інформації методом «Move to Front” та подання натуральних чисел методом Левенштейна

Мета: Набути практичні навички застосування методів кодування Move to Front («стопки книг») та кодування додатних цілих чисел методом Левенштейна.

Обладнання та програмне забезпечення: персональний комп’ютер; будь-яка мова програмування; офісне програмне забезпечення для формування звітів та побудови діаграм.

Література

1. Соколов А. Теорія інформації та кодування. Лабораторний практикум [Електронний ресурс]: режим доступу: https://books.google.com.ua/books?id=XQRPDwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false
2. Move-to-front transform - Wikipedia
3. Levenshtein coding - Wikipedia
4. Євсеєв С.П. Кібербезпека: Основи кодування та криптографії / Євсеєв С.П., Мілов О.В., Остапов С.Е. Сєвєрінов О.В. - Харків, - Львів: Вид. ПП “Новий світ-2000”, 2023. - 658 с.
5. Теорія інформації і кодування: курс лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за спеціальністю 124 «Системний аналіз» /; уклад.: А.Є.Коваленко. Київ : КПІ ім. Ігоря Сікорського, 2020. - 248 с.
6. Івашко А.В., Крилова В.А. Теорія інформації та кодування в прикладах і задачах: навч.-метод. посіб. Харків : НТУ «ХПІ», 2022. - 317 с.
7. Жураковський Ю.П., Полторак В.П. Теорія інформації та кодування: Підручник. – К.: Вища шк., 2001. – 255 с.

Теоретична частина

Метод кодування Move to Front (“стопки книг”) було запропоновано у 1980 році. Він ґрунтується на частотних характеристиках мови, тобто літери, що входять у фразу, яка кодується.

Починають кодування зі стартового стану.

Стартовий стан: літери абетки кодуються десятковим числом, яке вказує положення літери в оригінальній абетці, починаючи з нуля. Наприклад, в українській абетці: а — 0; б — 1; в — 2 і так далі.

Припустимо, необхідно закодувати вираз «стопка книг». Стартовий стан буде таким:

а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Кодуємо першу літеру виразу, «с». Її код у таблиці — 20. Переставляємо закодовану літеру на перше місце. Отримаємо таку таблицю:

с	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Кодуємо другу літеру, «т» - 21. Отримаємо таку табличку:

т	с	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Третя літера, «о» буде мати код 19. Переставляємо її на першу позицію. Отримаємо таку табличку:

о	т	с	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	п	р	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Літера «п» - 20. Результуюча табличка:

п	о	т	с	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	р	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Літера «к» - 17. Таблиця:

к	п	о	т	с	а	б	в	г	д	е	є	ж	з	и	і	ї	й	л	м	н	р	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Літера «а» - 5. Таблиця:

а	к	п	о	т	с	б	в	г	д	е	є	ж	з	и	і	ї	й	л	м	н	р	у	ф	х	ц	ч	ш	щ	ь	ю	я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Таким чином, код першого слова «стопка» буде виглядати так: 20, 21,19,20,17,5.

Аналогічно кодується й друге слово виразу.

Результат: 20, 21,19,20,17,5,1,20,15,10.

Перевагою цього методу кодування буде те, що літери, які частіше зустрічаються у тексті (при достатньому обсязі тексту), будуть мати менший код, ніж при звичайному кодуванні, як у стартовій таблиці. У нашому прикладі цього не видно, але звісно, у такому короткому фрагменті не можна чекати якихось статистичних особливостей.

Розкодування виконується так само. Береться стартова табличка і знаходиться літера під номером 20, яку потім переставляють на перше місце. Далі — літера під номером 21 і так далі.

Кодування додатних цілих чисел методом Левенштейна

Отриманий код методом “стопка книг” можна додатково закодувати методом Левенштейна.

Співвідношення між вхідним та закодованим словом кода Левенштейна визначається таким чином:

Крок 1. Ініціалізується лічильник кроків *C*=1, *K* — код числа (спочатку порожній).

Крок 2. Записують двійковий код числа, що кодується, без старшої “1”, наприклад, число 1100 записується як 100, а число 100 — як 00.

Крок 3. Дописують отриманий фрагмент кода на початку *K*.

Крок 4. Нехай *M* — кількість бітів, записаних на кроці 2 у двійковому форматі.

Крок 5. Якщо *M* не порожньо, то *C*=*C*+1, а кроки 2-4 повторюються для отриманого *M*. Інакше переходять на крок 6.

Крок 6. Записують *C* одиниць і 0 на початку коду *K* (наприклад, якщо лічильник *C*=2, *K*=0 011, то результат: *K*=110 0 011) — код Левенштейна.

Співвідношення між закодованим та вхідним словом коду Левенштейна визначається таким способом:

Крок 1. Порахувати кількість *C* одиничних бітів до першого нульового біта.

Крок 2. Якщо *C*=0, то закодоване значення — 0. Якщо ні, перейти на крок 3.

Крок 3. Відкинути з *K* перші *C* одиниць і наступний за ними 0. Записати нове значення *K*.

Крок 4. Встановити змінну *N*=1. Увести лічільник кроків *P*=*C*-1.

Крок 5. Якщо *P*=0, то *N* — шукане число. Якщо ні — перейти на крок 6.

Крок 6. Прочитати перші *N* бітів з *K*. Записати нове значення *K* без цих *N* бітів.

Крок 7. До прочитаних бітів додати 1 в початок (наприклад, прочитано 00, отримаємо 100).

Крок 8. Перевести отримане значення до десяткової системи числення. Це буде нове значення *N* (при умові, що як вхідна використовувалася саме 10-кова система числення).

Крок 9. *P*=*P*-1. Повторити, починаючи з кроку 5.

Для довідки подаємо табличку перших 24 чисел, закодованих методом Левенштейна [3]:

```
 0 0
 1 10
 2 110 0
 3 110 1
 4 1110 0 00
 5 1110 0 01
 6 1110 0 10
 7 1110 0 11
 8 1110 1 000
 9 1110 1 001
10 1110 1 010
11 1110 1 011
12 1110 1 100
13 1110 1 101
14 1110 1 110
15 1110 1 111
16 11110 0 00 0000
17 11110 0 00 0001
18 11110 0 00 0010
19 11110 0 00 0011
20 11110 0 00 0100
21 11110 0 00 0101
22 11110 0 00 0110
23 11110 0 00 0111
24 11110 0 00 1000
```

Коефіцієнт стиску:

$$K_P = \frac{\log_2 m \times N}{N_1};$$

де m — потужність вхідного алфавіту; N — довжина вхідного повідомлення; N_1 — довжина закодованого повідомлення.

Практична частина

Для виконання цієї лабораторної роботи необхідно зробити таке:

Крок 1. Складіть програму (будь-якою мовою програмування), яка кодує довільне вхідне повідомлення методом *Move to Front* (“стопки книг”) та виводить результат на екран/файл.

Крок 2. Складіть програму (або модифікуйте попередню), яка кодує методом Левенштейна результат, отриманий методом *Move to Front*. Обидві програми мають як кодувати, так і успішно розкодовувати вхідне повідомлення.

Крок 3. Обчисліть коефіцієнти стиску в обох випадках. Поясніть отримані результати.

Крок 4. Підготуйте звіт з лабораторної роботи, до якого включіть:

- Поясніть алгоритм *Move to Front*, його переваги та недоліки. Наведіть приклади на основі Ваших обчислень у цій лабораторній роботі.
- Поясніть алгоритм Левенштейна, його переваги та недоліки на основі Ваших обчислень.
- відповіді на контрольні запитання.

Контрольні запитання

1. Поясніть область використання алгоритму Move to Front.
2. Які області використання алгоритму Левенштейна?
3. Які переваги та недоліки вивчених алгоритмів Ви бачите?
4. Для яких типів вхідних даних використовується алгоритм Левенштейна?